CS 280 - Homework 3
Due: Friday, September 26

In the following, when asked about algorithms, you should use the notation from the text (chapter 1) and estimate their running time.

1. Construct and analyse an algorithm to solve Problem 5.2 from Homework 2.

2. A recursive version of insertion sort could be expressed as sorting an array of length n by sorting the first $(n-1)$ terms and then inserting the n-th term. Construct and analyse this algorithm.

3. If $A[n]$ is an array of n distinct numbers, then we define a pair (i, j) to be an "inversion" of A if $(i < j)$ and $(A[i] > A[j])$.

i. Give an array with numbers from the set $\{1, 2, ..., n\}$ having the most inversions and state how many inversions it has.
ii. What is the relationship between the running time of the usual form of insertion sort and the number of inversions in the input array?
iii. Construct and analyse an algorithm to determine the number of inversions in any permutation on n elements in $O(n \log n)$ worst-case time.

The following are induction proof questions.

4. Prove each of the following arithmetic statements by induction:

i. Question 2.2.16 from the textbook; i.e., discover and prove a formula for the n-th power of the 2x2 matrix:
$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$
ii. $(n-r)! \cdot r!$ divides $n!$ for all n and for all r (with $0 \leq r \leq n$).
iii. $(a+b)^n = a^n + n \cdot a^{(n-1)} \cdot b + \cdots + nCr \cdot a^{(n-r)} \cdot b^r + \cdots + nCn \cdot b^n$, where
$$nCr := \frac{n!}{(n-r!) \cdot r!}$$
Hint: show first that $nCr = nC(n-r)$ and $nCr + nC(r-1) = (n+1)Cr$.
iv. $nC0 + nC1 + ... + nCn = 2^n$.

5. Devise, state clearly, and prove by induction, a result analogous to the Euclidean algorithm for polynomials in one real variable and with real coefficients.